Amendments to the Specification:

Please amend the following paragraph beginning at page 5, line 11 as follows:

As seen in Figure 1, the preferred embodiment of four processors 107, 108, 109, 110, connected by interprocessor high-speed interconnect 111, are assigned the tasks of handling network protocol, but any number of processors can be assigned to handle network protocol tasks, depending on the capabilities of the processors and of the complexity of the network protocol. Fibre Channel input traffic is handled by FC Inbound Processor 107, traffic into the Infiniband network is handled by the IB ~~Inbound~~ Outbound Processor ~~108~~ 109, outbound traffic into the Fibre Channel network is handled by FC Outbound Processor 110, and traffic coming from the Infiniband network is handled by IB ~~outbound~~ inbound processor ~~109~~ 108. Within ~~a~~ each processor 107, 108, 109, or 110, several identical threads perform various protocol sub-tasks.

Please amend the following paragraph beginning at page 6, line 14 as follows:

Referring back to Figure 1, inbound frames are first stored in frame first-in, first-out registers (FIFOs) at the FC interface 105. These can be dual FIFOs that operate in ping-pong fashion to prevent frame overruns or can be implemented as a single FIFO. In the preferred embodiment, associated with the FIFOs are two addresses, one pointing to a memory area in the FC inbound processor 107 where the header of an incoming frame will be written into, and the other pointing to a memory area in the IB ~~inbound~~ outbound processor ~~108~~ 109, for writing the payload of the frame (i.e., the data). Or in another embodiment, only one address is associated with the FIFO block, storing both the header and the payload of the frame. These addresses are set by the FC inbound processor 107 that manages the respective memory areas. Once a frame begins arriving into the FIFO, the FIFO logic moves first the header and then the payload into the preassigned inbound data block (IBDB) areas over the highspeed interconnection device.

Please amend the following paragraph beginning at page 8, line 19 as follows:

All input queues in this architecture are organized as ~~a~~ ring buffers, with the first and last valid buffer locations recorded in the head and tail pointers 501 and 502 of the

particular queue, respectively, as shown in Figure 5. The head and tail pointers for each queue can be implemented as dedicated registers in hardware, or can be stored in the memory on the previously determined locations. In the latter case, the addresses of the memory locations containing these pointers are determined before the processing has begun and are not changed during the processing. In addition, the queues themselves can be implemented as dedicated hardware or can be stored in the memory on the previously determined locations.

Please amend the following paragraph beginning at page 11, line 9 as follows:

If the frame requires a response to be generated to the sender as determined in decision block 7 10, such as acknowledgment or busy message, a response frame is constructed in function block 711. All required data for the response are collected in an OBDB (outbound data block) block and the pointer of the OBDB block is sent to the outbound processor 110 (Figure 1), which composes and transfers the response frame. The FC outbound processor 110 composes the response frame and transfers it to the outbound FIFO in the FC interface 106 in Figure 1. The IBDB address of the data frame is then placed in the work queue of the thread for transferring the data to the host interface FIFO 112 (Figure 1), as shown in the function block 712. This task is performed by the Infiniband ~~inbound~~ outbound processor ~~108~~ 109 (Figure 1). From here, the data are sent to the Infiniband network.

Please amend the following paragraph beginning at page 11, line 26 as follows:

In the FC inbound processor 107 (Figure 1) there is a master thread that assigns the processing of incoming frame headers to protocol threads. This is done by the master thread placing requests in queues stored in memory. When these queues become large, indicating that the processing of the frames has slowed down, the master thread can create request service queues in another processor's memory (such as the IB ~~inbound~~ outbound processor ~~108~~ 109), thus having threads in that processor service new incoming frames. This will reduce the workload in the FC inbound processor, thus increasing protocol handler performance.

Please amend the following paragraph beginning at page 12, line 8 as follows:

The process is shown in Figure 8 and begins when a frame is received at the interface in function block 801. When this occurs, the inbound processor is notified and, in response, the inbound processor checks the size of its work queue in function block 802. A determination is made in decision block 803 as to whether the work queue is above a predetermined threshold. If not, the pointer to the inbound frame is placed on the work queue in function block 804, and the process loops back to function block 801 to await the next frame. If, however, the threshold is exceeded, ~~th~~ the inbound processor checks the size of another processor's work queue in function block 805. A determination is made in decision block 806 as to whether the work queue of the other processor is above the threshold. If not, a pointer to the inbound frame is placed on the work queue of the other processor in function block 807, and the process loops back to function block 801 to await the next frame. If, however, the threshold is exceeded in the other processor's work queue, a determination is made in decision block 808 as to whether there are any other processors available. If so, the process loops back to function block 805 to check that processor's work queue; otherwise, the processor indicates to the interface to stop flow of incoming frames via pacing in function block 809. The process then loops back to function block 802 where the inbound processor again checks the size of its work queue.

Please amend the following paragraph beginning at page 13, line 2 as follows:

Incoming frames are separated into header and payload parts, and the header is stored in the memory of the FC inbound processor 107 while the payload is stored in the memory of the IB ~~inbound~~ outbound processor 108, so that it can be properly formatted for delivery to a destination in the Infiniband network. Associated with each processor's memory is a memory free list that provides pointers to available memory blocks. If the free list associated with a processor's memory is about to become empty, indicating that there is not any more available memory for storing incoming frames in that processor, the master thread that assigns memory blocks to the incoming frames can examine the free lists in other processors to determine the availability of corresponding memory. Then, the

master thread can redirect the incoming traffic to a ~~process~~ processor whose memory is under utilized.